The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A computer-implemented method for ~~substantially eliminating~~ **reducing** C recursion from the execution of static initializer methods in a virtual machine environment, the method comprising:

rewriting native C code associated with a static initializer as a ~~Java™~~ **JAVA programming language** method;

using a transition frame in a ~~Java™~~ **JAVA programming language** stack to execute the ~~Java™~~ method;

using a native method to manipulate the ~~Java™~~ **JAVA programming language** stack; and

using a first opcode in the transition frame.

2. (original) A method as recited in claim 1 wherein using the first opcode in the transition frame includes using the first opcode to determine that the transition frame is associated with the static initializer.

3. (original) A method as recited in claim 2 further including causing the static initializer to run, wherein the static initializer is caused to run by a second opcode.

4. (original) A method as recited in claim 3 further including resuming execution at the second opcode after the static initializer has run.

5. (cancelled)

6. (original) A method as recited in claim 1 wherein the native C code includes code for identifying the static initializer.

7. (currently amended) An apparatus for ~~substantially eliminating~~ **reducing** C recursion from the execution of static initializer methods in a virtual machine environment, the method comprising:

a means for rewriting native C code associated with a static initializer as a ~~Java™~~ **JAVA programming language** method;

a means for using a transition frame in a ~~Java™~~ **JAVA programming language** stack to execute the ~~Java™~~ **JAVA programming language** method;

a means for using a native method to manipulate the ~~Java™~~ **JAVA programming language** stack; and
a means for using a first opcode in the transition frame.

8. (original) An apparatus as recited in claim 7 further comprising:
a means for using the first opcode to determine that the transition frame is associated with the static initializer.

9. (original) An apparatus as recited in claim 8 further comprising:
a means for causing the static initializer to run, wherein the static initializer is caused to run by a second opcode.

10. (original) An apparatus as recited in claim 9 further comprising:
a means for resuming execution at the second opcode after the static initializer has run.

11. (original) An apparatus as recited in claim 7 wherein the native C code includes code for identifying the static initializer.

12. (currently amended) A computer program product for substantially eliminating C recursion from the execution of static initializer methods in a virtual machine environment, comprising:

computer code for rewriting native C code associated with a static initializer as a ~~Java™~~ **JAVA programming language** method;

computer code for using a transition frame in a ~~Java™~~ **JAVA programming language** stack to execute the ~~Java™~~ **JAVA programming language** method;

computer code for using a native method to manipulate the ~~Java™~~ **JAVA programming language** stack;

computer code for using a first opcode in the transition frame; and

a computer readable medium for storing the computer program product.

13. (previsouly presented) A computer program product as recited in claim 12 wherein using the first opcode in the transition frame includes using the first opcode to determine that the transition frame is associated with the static initializer.

14. (previsouly presented) A computer program product as recited in claim 13 further including:

computer code for causing the static initializer to run, wherein the static initializer is caused to run by a second opcode.

15. (previsouly presented) A computer program product as recited in claim 14 further including:

computer code for resuming execution at the second opcode after the static initializer has run.

16. (cancelled)

17. (previsouly presented) A computer program product as recited in claim 12 wherein the native C code includes code for identifying the static initializer.